

IN THE CLAIMS:

A status of all the claims of the present Application is presented below:

1. **(Currently amended)** A lexical search tree data structure accessible by a computer, comprising:

a plurality of linked root nodes;

at least one branch linked to at least one of said plurality of root nodes, each branch along with the root node to which it is linked representing at least one of a plurality of signatures, a first character of each signature being represented by one of said plurality of root nodes, each one of said plurality of root nodes comprising a hash value for the first character represented by said respective root node; and

each branch having one or more leaf nodes linked hierarchically to one another, each leaf node representing a character in a signature.

2. **(Original)** The lexical search tree data structure of claim 1, further comprising a twig linked to one of said leaf nodes and representing a substring of a second signature of said plurality of signatures, said second signature having at least the same first character as said first signature and said first and second signatures diverging from one another at said leaf node to which said twig is linked.

3. **(Original)** The lexical search tree data structure of claim 2, said twig comprising:

a twig node representing a first character of said substring, said twig node being at the same level as said leaf node to which said twig is linked; and

one or more leaf nodes, each leaf node representing a character of said substring.

4. **(Original)** The lexical search tree data structure of claim 1, wherein each of said plurality of signatures comprises a string of characters.

5. **(Original)** The lexical search tree data structure of claim 1, wherein the number of said root nodes is equal to the number of characters in a character set available to represent said plurality of signatures.

6. **(Original)** The lexical search tree data structure of claim 5, wherein said character set comprises the set of ASCII characters.

7. **(Canceled)**

8. **(Currently amended)** The lexical search tree data structure of claim [[7]] 1, each root node further comprising a pointer to a leaf node of said one or more leaf nodes if a first character of any of said plurality of signatures corresponds to said root node.

9. **(Original)** The lexical search tree data structure of claim 1, each leaf node having only one other leaf node directly linked to it at the next lower level.

10. **(Original)** The lexical search tree data structure of claim 1, further comprising a plurality of twigs linked to one of said leaf nodes, each twig of said plurality of twigs representing a substring of a different signature of said plurality of signatures.

11. **(Currently amended)** A method for searching a plurality of signatures stored in a lexical search tree data structure ~~by a computer~~, said method comprising:

determining a hash value for a target signature;

determining a branch ~~associated with~~ having a root node of said lexical search tree data structure corresponding to said hash value, said branch along with said root node representing at least one signature of said plurality of signatures, said branch having one or more leaf nodes linked hierarchically to one another, each leaf node representing an element of said at least one signature; and

traversing only said branch to find a match between said at least one signature and said target signature.

12. **(Original)** The method of claim 11, said determining a hash value comprising:

determining a first element of said target signature; and
determining a hash value for said first element.

13. **(Original)** The method of claim 12, said hash value being the ASCII code for said first element.

14. **(Original)** The method of claim 11, said traversing only said branch comprising comparing successive elements of said target signature with successive elements of said at least one signature stored in successive leaf nodes of said one or more leaf nodes so long as said successive elements of said target signature match said successive elements of said at least one signature.

15. **(Original)** The method of claim 11, said traversing only said branch further comprising:

determining a twig associated with said branch at a point of divergence between said at least one signature and said target signature, said twig representing a terminating substring of a second signature of said plurality of signatures; and

traversing said twig to find a match between a terminating substring of said target signature and said terminating substring represented by said twig.

16. **(Original)** The method of claim 15, said traversing said twig comprising comparing successive elements of said terminating substring of said target signature with successive elements of said terminating substring of said second signature represented by said twig so long as said successive elements match.

17. **(Original)** The method of claim 14, said traversing only said branch further comprising:

setting a current node pointer to point to a leaf node of said one or more leaf nodes;
setting a target signature pointer to point to an element of said target signature;
in response to a value of said leaf node pointed to by said current node pointer being equal to a wild card character and a value of the element pointed to by said target signature pointer being equal to a value of the next leaf node following the leaf node pointed to by said current node pointer, updating said current node pointer to point to a leaf node following said next leaf node.

18. **(Currently amended)** A method for representing a plurality of signatures in a lexical search tree data structure accessibly by a computer, comprising:

- a) allocating a plurality of root nodes, one for each distinct element of said plurality of signatures;
- b) determining ~~an index a hash~~ value for a signature of said plurality of signatures;
- c) determining a status of a root node ~~corresponding to having~~ said determined ~~index hash~~ value, said root node being selected from said plurality of root nodes and representing a first element of said signature;
- d) creating a branch for said root node if said root node has no existing branch, said branch having one or more leaf nodes linked hierarchically to one another, each successive leaf node representing a successive element of said signature;
- e) creating a twig for said root node if said root node has an existing branch, said twig linked to one of said leaf nodes and representing a substring of said signature, the first element of said substring being represented by a twig node linked to said one of said leaf nodes; and
- f) repeating steps (b) through (e) for each signature of said plurality of signatures.

19. **(Original)** The method of claim 18, said determining index value comprising:
determining a first element of said signature; and
determining an ASCII code for said first element.

20. **(Original)** The method of claim 18, said creating said twig comprising:
determining the location of said one of said leaf nodes from which said twig diverges.